OIPE IAP51
SEP 28 2005
PATENT & TRADEMARK OFFICE

09-29-05

EV697604618

IFW
AF
2194

PTO/SB/21 (08-03)
Approved for use through 07/31/2006. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Total Number of Pages in This Submission

Application Number	09/884,374
Filing Date	6/18/2001
First Named Inventor	Bor-Ming Hsieh
Group Art Unit	2194
Examiner Name	QING YUAN WU
Attorney Docket Number	MS1-742US

ENCLOSURES (check all that apply)

- ☒ Fee Transmittal Form
  - ☐ Fee Attached
- ☐ Amendment / Reply
  - ☐ After Final
  - ☐ Affidavits/declaration(s)
- ☐ Extension of Time Request
- ☐ Express Abandonment Request
- ☐ Information Disclosure Statement
- ☐ Certified Copy of Priority Documents
- ☐ Response to Missing Parts/ Incomplete Application
  - ☐ Response to Missing Parts under 37 CFR 1.52 or 1.53
- ☐ Drawing(s)
- ☐ Licensing-related Papers
- ☐ Petition
- ☐ Petition to Convert to a Provisional Application
- ☐ Power of Attorney, Revocation Change of Correspondence Address
- ☐ Terminal Disclaimer
- ☐ Request for Refund
- ☐ CD, Number of CD(s)
- ☐ After Allowance Communication to Group
- ☐ Appeal Communication to Board of Appeals and Interferences
- ☒ Appeal Communication to Group ***(Appeal Notice, Brief, Reply Brief)***
- ☐ Proprietary Information
- ☐ Status Letter
- ☒ Other Enclosure(s) *(please identify below):*

Appeal Brief (47 pages); return receipt postcard

Remarks

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm or Individual Name	**Brian G. Hart/Reg. No. 44421**
Signature	Brian Hart
Date	**September 28, 2005**

CERTIFICATE OF TRANSMISSION/ EXPRESS MAILING

I hereby certify that this correspondence is being facsimile transmitted to the USPTO or deposited with the United States Postal Service with sufficient postage as EXPRESS mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on the date shown below.

Typed or printed name	**Cheryl Boies**		
Signature	Cheryl Boies	Date	9-28-05

This collection of information is required by 37 CFR 1.[illegible]. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

OIPE
SEP 2 8 2005
IAP51
PATENT & TRADEMARK OFFICE

EV697604618

PTO/SB/17 (12-04)
Approved for use through 07/31/2006. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
...aperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

Effective on 12/08/2004.
Fees pursuant to the Consolidated Appropriations Act, 2005 (H.R. 4818).

FEE TRANSMITTAL
For FY 2005

☐ Applicant claims small entity status. See 37 CFR 1.27

TOTAL AMOUNT OF PAYMENT	($) 500.00

Complete if Known	
Application Number	09/884,374
Filing Date	6/18/2001
First Named Inventor	Bor-Ming Hsieh
Examiner Name	QING YUAN WU
Art Unit	2194
Attorney Docket No.	MS1 -742US

METHOD OF PAYMENT (check all that apply)

☐ Check ☐ Credit Card ☐ Money Order ☐ None ☐ Other (please identify): _____

☑ Deposit Account Deposit Account Number: 12-0769 Deposit Account Name: Lee & Hayes, PLLC

For the above-identified deposit account, the Director is hereby authorized to: (check all that apply)

☑ Charge fee(s) indicated below ☐ Charge fee(s) indicated below, **except for the filing fee**

☑ Charge any additional fee(s) or underpayments of fee(s) under 37 CFR 1.16 and 1.17 ☑ Credit any overpayments

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

FEE CALCULATION

1. BASIC FILING, SEARCH, AND EXAMINATION FEES

	FILING FEES		SEARCH FEES		EXAMINATION FEES		
Application Type	Fee ($)	Small Entity Fee ($)	Fee ($)	Small Entity Fee ($)	Fee ($)	Small Entity Fee ($)	Fees Paid ($)
Utility	300	150	500	250	200	100	
Design	200	100	100	50	130	65	
Plant	200	100	300	150	160	80	
Reissue	300	150	500	250	600	300	
Provisional	200	100	0	0	0	0	

2. EXCESS CLAIM FEES

Fee Description	Fee ($)	Small Entity Fee ($)
Each claim over 20 or, for Reissues, each claim over 20 and more than in the original patent	50	25
Each independent claim over 3 or, for Reissues, each independent claim more than in the original patent	200	100
Multiple dependent claims	360	180

Total Claims	Extra Claims	Fee ($)	Fee Paid ($)
_____ - 20 or HP =	_____	x 50	= _____

HP = highest number of total claims paid for, if greater than 20

Indep. Claims	Extra Claims	Fee ($)	Fee Paid ($)
_____ - 3 or HP =	_____	x 200	= _____

HP = highest number of independent claims paid for, if greater than 3

Multiple Dependent Claims	
Fee ($)	Fee Paid ($)
_____	_____

3. APPLICATION SIZE FEE

If the specification and drawings exceed 100 sheets of paper, the application size fee due is $250 ($125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).

Total Sheets	Extra Sheets	Number of each additional 50 or fraction thereof	Fee ($)	Fee Paid ($)
_____ - 100 =	_____	/ 50 = _____ (round up to a whole number)	x _____	= _____

4. OTHER FEE(S)

	Fees Paid ($)
Non-English Specification, $130 fee (no small entity discount)	
Other: Appeal Brief	$500.00

SUBMITTED BY

Signature	Brian Hart	Registration No. (Attorney/Agent) 44421	Telephone (509) 324-9256
Name (Print/Type)	Brian G. Hart		Date 9/28/2005

This collection of information is required by 37 CFR 1.136. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 30 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

OIPE
SEP 2 8 2005
IAP51
PATENT & TRADEMARK OFFICE

EV697604618

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Application Serial No. 09/884,374
Filing Date ..... Jun 18, 2001
Inventorship .....Hsieh
Appellant..... Microsoft Corporation
Group Art Unit.....2127
Examiner .....Wu, Qing-Yuan
Attorney's Docket No. MS1-742US
Title: Sleep Queue Management

APPEAL BRIEF RESPONSIVE TO THE JUNE 13, 2005 FINAL OFFICE ACTION

To: To:Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-14503

From: Brian G. Hart (Tel. 509-324-9256; Fax 509.323-8979)
Lee & Hayes, PLLC
421 W. Riverside Avenue, Suite 500
Spokane, WA 99201

09/30/2005 AWONDAF1 00000052 120769 09884374
01 FC:1402 500.00 DA

Introduction

Pursuant to 37 C.F.R. § 41.37, Appellant hereby submits an appeal brief within two months of the requisite time from the date of filing the Notice of Appeal, which was filed on August 1, 2005. Appellant appeals to the Board of Patent Appeals and Interferences seeking review of rejections to subject matter in the above captioned patent application. These rejections result from a Final Office Action dated June 13, 2005 (hereinafter referred to as the "Final Action").

REAL PARTY IN INTEREST

The real party in interest is Microsoft Corporation, the assignee of all right, title and interest in and to the claimed subject matter.

RELATED APPEALS AND INTERFERENCES

Appellant is not aware of any other appeals, interferences, or judicial proceedings that will directly affect, be directly affected by, or otherwise have a bearing on the Board's decision to this pending appeal.

STATUS OF CLAIMS

Claims 1-56 were originally pending.

Claims 1-7, 9, 11-13, 15-16, 18, 20-30, 37, 39, 41-42, 46, and 48-49 were amended.

Claims 8, 10, 19, 38, 45, 47, and 50-56 were canceled without prejudice.

No claims were added.

Accordingly, claims 1-7, 9, 11-18, 20-37, 39-44, 46, and 48-49 remaining pending.

Claims 1-7, 9, 11-18, 20-37, 39-44, 46, and 48-49 stand rejected. Appellant appeals the rejections of pending claims 1-7, 9, 11-18, 20-37, 39-44, 46, and 48-49. These pending claims are presented in the Appendix of Appealed Claims on page 28.

STATUS OF AMENDMENTS

Appellant has not amended the claims subsequent to the date of the Final Action.

SUMMARY OF CLAIMED SUBJECT MATTER

This summary section provides a concise explanation of each of the independent claims, including specific reference characters and reference to the specification. These specific reference characters are examples of particular elements of the drawings for certain claimed embodiments. It is understood that the claims are not to be limited to solely the elements corresponding to these reference characters and that this section is provided to comply with the requirement of 37 CFR § 41.37(c)(1)(v).

Independent **claim 1** recites in part "a computer implemented method for managing a multi-dimensional sleep queue". An exemplary "multi-dimensional sleep queue" 300 is described pages 8 through 13 in the specification with respect to Fig. 3. An exemplary "computer implemented method for managing the multidimensional sleep queue", as claim 1 recites, is described in pages 13 through 20 of the specification with respect to Figs. 4 through 9. An exemplary computing device for the "computer implemented method" is described on pages 25 through 31 of the specification with respect to Fig. 14.

Independent claim 1 further recites "identifying a thread of execution to insert into a sleep queue for a predetermined amount of time", and "responsive to the identifying, inserting the thread of execution into a first dimension of the

multi-dimensional sleep queue if: (a) there is not a thread with a wake-up time equivalent to the predetermined amount of time in the first dimension; and (b) if there are one or more different threads of execution with the wake-up time in a second dimension of the multi-dimensional sleep queue, each of the one [or] more different threads of execution has a thread priority lower than or equal to a thread priority associated with the thread of execution." Various embodiments of these claimed features are described in the specification at pages 13-20 with respect to Figs. 4-9.

Independent **claim 13** recites "identifying a thread of execution to insert into a sleep queue for a predetermined amount of time", "responsive to the identifying, inserting the thread of execution into a first dimension of the multi-dimensional sleep queue if: (a) there is not a thread with a wake-up time equivalent to the predetermined amount of time in the first dimension; and (b) if there are one or more different threads of execution with the wake-up time in a second dimension of the multi-dimensional sleep queue, each of the one [or] more different threads of execution has a thread priority lower than or equal to a thread priority associated with the thread of execution." These claimed features are described in the specification with respect to Figs. 4 and 5 (please see pages 13 through 15), according to one embodiment. Additionally, these claimed features are described in the specification with respect to Figs. 6 through 9 (please see pages 15 through 20), according to another embodiment.

Independent **claim 24** recites "[a] computer implemented method for managing a multi-dimensional sleep queue". An exemplary multi-dimensional sleep queue 300 is described with respect to Fig. 3 in the specification at pages 8

through 13. An exemplary computing device for implementing the method to manage the multi-dimensional sleep queue is described in the specification with respect to Fig. 14 at pages 25 through 31. Independent claim 24 further recites "inserting a new thread into the multi-dimensional sleep queue using a multi-dimensional atomic walk procedure", and "removing the new thread from the multi-dimensional sleep queue for insertion into a run queue." These claimed features are described in the specification with respect to Figs. 6 through 9 at pages 15 through 20.

Independent **claim 31** recites "[a] computer-readable medium comprising computer-executable instructions for managing a sleep queue". A computer-readable medium comprising computer-executable instructions is described in the specification with respect to Fig. 14 at pages 26 through 31. An exemplary multi-dimensional sleep queue 300 is described in the specification with respect to Fig. 3 at pages 8 through 13. Claim 31 further recites "inserting a new thread into the sleep queue using a multi-dimensional atomic walk procedure", and "removing the new thread from the sleep queue for insertion into a run queue." These claimed features are described in the specification with respect to figures 6 through 9 at pages 15 through 20.

Independent **claim 39** recites "[a] system for managing a sleep queue", "a memory comprising computer-executable instructions and a multi-dimensional sleep queue", and "a processor operatively coupled to the memory for executing the computer-executable instructions". Such a system (1400), memory (1414), computer-executable instructions (1440) are described in the specification with respect to Fig. 14 at pages 25-31. "[A] multi-dimensional sleep queue", as claim

39 recites, is shown and described in the specification with respect to Fig. 3 at pages 8 through 13.

Independent claim 39 also recites "identifying a thread of execution to insert into a sleep queue for a predetermined amount of time", and "responsive to the identifying, inserting the thread of execution into a first dimension of the multi-dimensional sleep queue if: (a) there is not a thread with a wake-up time equivalent to the predetermined amount of time in the first dimension; and (b) if there are one or more different threads of execution with the wake-up time in a second dimension of the multi-dimensional sleep queue, each of the one [or] more different threads of execution has a thread priority lower than or equal to a thread priority associated with the thread of execution." These claimed operations are described in the specification with respect to Figs. 4 through 9 at 13 through 20.

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

The following grounds of rejection are for review on appeal:

- Claims 1-3, 5-7, 9, 11-18, 20-37, 39-44, 46, and 40 8-49 stand rejected by the Final Action under 35 USC §112, second paragraph, as being indefinite for failing to particularly pointed out and distinctly claim the subject matter which applicant regards as the invention.
- Claims 1-7, 9, 11-18, 20-37, 39-44, 46, and 48-49 stand rejected by the Final Action under 35 USC §103(a) as being unpatentable over Applicant Admitted Prior Art (hereinafter AAPA), in view of US patent 6,609,161 to Young.

ARGUMENT

35 USC §112, Second Paragraph Rejections

Claims 1-3, 5-7, 9, 11-18, 20-37, 39-44, 46, and 40 8-49 stand rejected under 35 USC §112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which Appellant regards as the invention. Appellant respectfully traverses these rejections.

Claim 1 recites in part (emphasis added) "identifying a thread of execution to insert into a sleep queue for **a predetermined amount of time**", and "responsive to the identifying, inserting the thread of execution into a first dimension of the multi-dimensional sleep queue if: (a) there is not a thread with **a wake-up time equivalent to the predetermined amount of time** in the first dimension; and (b) if there are one or more different threads of execution with **the wake-up time** in a second dimension of the multi-dimensional sleep queue, each of the one [or] more different threads of execution has a thread priority lower than or equal to a thread priority associated with the thread of execution."

In addressing claim 1, the Final Action asserts that it is uncertain whether "the wake-up time" in lines 20-21 refers to the preceding "a wake-up time", which claim 1 indicates is "equivalent to the predetermined amount of time". Appellant respectfully disagrees. Terms in claim 1 properly refer to their antecedent. Specifically, the claimed "a wake-up time equivalent to the predetermined amount of time" provides proper antecedent basis for the subsequently recited feature of "the wake-up time". (The phrase "a predetermined amount of time" provides antecedent basis for "the predetermined amount of time"). Thus, "the wake-up

time" properly refers to its antecedent "a wake-up time", which claim 1 also clearly indicates is "equivalent to the predetermined amount of time". As such, claim 1 particularly points out and distinctly claims the subject matter of the invention.

Accordingly, withdrawal of the 35 USC §112, second paragraph, rejection of claim 1 is respectfully requested.

Claims 13 and 39 are rejected by the Final Action under 35 USC §112, second paragraph, with the same rationale used to reject claim 1. For the reasons already discussed both respect to claim 1, claims 13 and 39 use consistent terms that properly refer to the antecedent. As a result, claims 13 and 39 particularly point out and distinctly claim subject matter which Appellant regards as the invention.

Accordingly, withdrawal of the 35 USC §112, second paragraph, rejection of claims 13 and 39 is respectfully requested.

The Final Action rejects **claims 2-3, 5-7, 9, 11-12, 14-18, 20-37, 40-44, 46, and 48-49** under 35 USC §112, second paragraph, without providing any reasoning to support the claim rejections. MPEP §707 states that "[i]n accordance with the patent statute, "[w]henever, on examination, any claim for a patent is rejected, [...] notification of the reasons for rejection and/or objection together with such information and references as may be useful in judging the propriety of continuing the prosecution (35 U.S.C. 132) should be given." Thus, when a claim is rejected, specific reasoning to support the claim rejection should be provided.

Appellant respectfully submits that such specific reasoning and rationale for a claim rejection not only allows Appellant to evaluate the rejections in view of

the claimed subject matter, but such reasoning would also provide a complete application file history and enhance the clarity of the prosecution history record. Since the Final Action does not provide any reasoning for rejecting claims 2-3, 5-7, 9, 11-12, 14-18, 20-37, 40-44, 46, and 48-49 under 35 USC §112, second paragraph, withdrawal of the 35 USC §112, second paragraph, rejections to these claims is respectfully requested.

If claims 2-3, 5-7, 9, 11-12, 14-18, 20-37, 40-44, 46, and 48-49 are again rejected under this same statutory basis, the Office is respectfully requested to seriously evaluate the patentability of each and every feature in these claims so that specific reasoning to support each claim rejection can be provided to Appellant.

35 USC §103(a) Rejections

Claims 1-7, 9, 11-18, 20-37, 39-44, 46, and 48-49 stand rejected by the Final Action under 35 USC §103(a) as being unpatentable over Appellant Admitted Prior Art (hereinafter AAPA), in view of US patent 6,609,161 to Young. Appellant respectfully traverses these rejections.

Claim 1 recites:

> *"[a] computer implemented method for managing a multi-dimensional sleep queue, the method comprising",*
>
> *"identifying a thread of execution to insert into a sleep queue for a predetermined amount of time", and*
>
> *"responsive to the identifying, inserting the thread of execution into a first dimension of the multi-dimensional sleep queue if:*

> *(a) there is not a thread with a wake-up time equivalent to the predetermined amount of time in the first dimension; and*
>
> *(b) if there are one or more different threads of execution with the wake-up time in a second dimension of the multi-dimensional sleep queue, each of the one [or] more different threads of execution has a thread priority lower than or equal to a thread priority associated with the thread of execution."*

In addressing claim 1, section 9 of the Final Action admits that AAPA does not "teach inserting the thread of execution into a first dimension of a multi-dimensional sleep queue if there is not a thread with a wake-up time equivalent to the predetermined amount of time in the first dimension." Attempting to show a suggestion for this admitted missing feature, the Action proposes a combination of AAPA and Young. However, the Action in section 11 admits that even this proposed combination of AAPA in view of Young also does not teach "inserting the thread of execution into a first dimension of the multi-dimensional sleep queue if: [...] (b) if there are one or more different threads of execution with the wake-up time in a second dimension of the multi-dimensional sleep queue, each of the one [or] more different threads of execution has a thread priority lower than or equal to a thread priority associated with the thread of execution", as claim 1 recites.

Attempting to provide this feature of claim 1 that is admitted as missing from AAPA and Young, the Final Action states the following in sections 11 and 12:

- "Young disclosed if a valid tail pointer exists for a target device (i.e. same wake-up time) appending the new SCB to the end of the target queue; in which SCB priorities are based on order of arrival [Young, col. 5, lines 52-60, col. 8, lines 31-34]."

- "AAPA disclosed sorting threads based on wake-up time and thread priority in a single-dimension sleep queue [AAPA page 3, lines 6-8]." Thus,
- "It would have been obvious to one of ordinary skill in the art, to have recognized that both the teaching of AAPA and Young execute/process threads/SCBs according to a priority scheduling semantic and the SCBs in the first dimension of Young's multi-dimensional queue have higher priority over the other SCBs in its corresponding second dimension target queue base of the priority scheduling semantic."

Appellant assumes that the Examiner wanted to modify AAPA and Young with the above quoted statements to arrive at the missing features of claim 1. However, Appellant respectfully submits that (with exception of the assertion that Young's target device represents a "same wake-up time") the Final Action's above statements regarding a "tail pointer", AAPA, and "a priority scheduling semantic", wherein SCB's in a first dimension have higher priority than SCBs in a second dimension, **do not modify AAPA in view of Young.** Since the Final Action admits that AAPA and Young do not teach all of the features of claim 1, and since the Final Action does not modify Young to arrive at these missing features, the Final Action has failed to present a prima facie case of obvious of claim 1 over AAPA and Young.

More particularly, the quoted statements of the Final Action in sections 11 and 12 merely indicate that AAPA sorts "threads based on wake-up time and thread priority in a single-dimension sleep queue" and Young "append[s] the new SCB to the end of the target queue; **in which SCB priorities are based on order of arrival**" – as described below, this is accomplished through use of the

mentioned "tail pointer". Appellant agrees with these two respective interpretations of AAPA and Young. Significantly, however, AAPA and Young's description of single dimension sorting and two-dimension order-of-arrival SCB storage does not teach or suggest those features of claim 1 that the Final Action admits are missing from AAPA and Young. AAPA never inserts anything into a second dimension. Moreover, as admitted by the Final Action (i.e., "appending the new SCB to the end of the target queue; in which SCB priorities are based on order of arrival"), the order-of-arrival scheduling semantic of Young always inserts an SCB for a target into a second dimension (not a first dimension) of a two-dimensional queue when there is already an SCB for that target in the second dimension. Thus, although claim 1 is rejected over AAPA and Young, the Final Action does not modify the combination to show that AAPA and Young teach or suggest those portions claim 1 that the Final Action admits are missing from AAPA and Young.

Appellant now describes AAPA and Young in greater detail to more clearly describe why the Examiner has failed to present a prima facie case of obvious of claim 1 over AAPA and Young.

AAPA describes prior art with respect to Figs. 1 and 2. Fig. 1 shows a conventional run queue. Page 1, line 16, through page 5, line 11, of AAPA describe conventional techniques to remove a thread from a conventional sleep queue and insert the thread into the conventional run queue. Fig. 2 of AAPA, which is described in Appellant's specification from page 3, line 3, through page 5, line 1, illustrates a conventional single dimension sleep queue. With respect to threads in a sleep queue, AAPA merely indicates that threads are stored in a

single-dimension sleep queue based on respective wake-up time and thread priority (i.e., page 3, lines 3-8). No techniques are described for inserting threads into a single-dimension sleep queue.

Young teaches a two-dimensional SCSI command block (SCB) execution queue to deliver multiple SCSI commands to a target such as a SCSI/peripheral device (please see the Abstract). (A SCSI command is not "a thread of execution", as claim one recites, but rather a command for a target device). Referring to Figs. 3A and 3B, and col. 4, lines 44-47, Young's two-dimensional SCB queue **260A** is a "common queue **275**" and "a plurality of target queues **270**A to **272**A". At col. 4, lines 63-65, Young describes that the "[c]ommon queue **275** includes one and only one SCB per target. Common queue **275** is a linked list of SCB's, i.e., a linked list and a first direction." "Each target queue is a linked list of SCB's for one specific target, i.e., a linked list in a second direction" (please see col. 5, lines 6-7). Thus, Young teaches a two-dimensional queue, with a first queue in a first direction to store "a maximum of one command block for a target" (column 2, lines 50-54) and one or more second queues in a second direction to store any additional command block(s) for a target that already has the maximum allowed one command block in the first queue.

At col. 3, lines 8-18 Young teaches how a command block directed to a target device that already has the one maximum allowed command block in the first queue **275** (described in the above paragraph) is added to the second queue (e.g., one of the target queues **270A** to **272A**). Specifically, Young teaches "[a] method of managing a command block execution queue where each command block is associated with one of a plurality of target devices includes testing a

predefined location in a list of target tail pointers to determine whether a target queue exists for a target device specified in a hardware command block. The method further includes **appending the hardware command block to a tail of the target queue for the target device upon the testing indicating that the target queue exists, and appending the hardware command block to a tail of a common queue upon the testing indicating that the target queue does not exist**." [Emphasis added].

Young teaches that operations to insert an SCB command into an SCB array are performed to maintain an order of arrival priority. Specifically, Young at column 6, lines 25-27 explicitly teaches "[u]sing method 400, SCBs are appended to the common and target queues **in the order of arrival** from host system 206" [emphasis added]. Additionally, at column 8, lines 25- 34, Young explicitly describes "[p]riority of SCB execution is **maintained according to the order of delivery by host system** 206, although commands for the same target are grouped together [...] Only tail pointers are required for the second dimension of queuing" [emphasis added]. It is well-known in the art that each SCSI command from an initiator to a target device is numbered with a command sequence number to ensure that every SCB is delivered to the target in the order that the command is transmitted. Thus, Young explicitly teaches that proper operation of the SCB array requires order-of-arrival SCB storage – as the Final Action admits (e.g., please see section 11)

With respect to use of a tail pointer to facilitate order-of-arrival SCB storage, Young explicitly teaches at column 8, lines 25- 34, that "[o]nly tail pointers are required for the second dimension of queuing." Young teaches at

column 5, lines 52 through column 6, lines 18, that the tail pointer of a second dimension queue is evaluated to determine whether a first control block for a particular target device already exists in the first dimension. If so, then any subsequent control blocks for that particular target device are always placed at the end of the queue in the second dimension – never in the first dimension. (The only time that a control block directed to a particular target is inserted into the first dimension of Young's queue is when no other control block directed to that particular target exists in the first dimension queue). Thus, Young's "tail pointer" (pointed out in the Final Action in section 11) merely allows Young to maintain order-of-arrival SCB storage in the SCB array.

The portions of Young cited by the Final Action (please see section 9) with respect to Fig. 3A, component **255**, Fig. 3B, component **260A**, Fig. 3C, Fig. 6, column 5, lines 46-51, and column 6, lines 1-7, maintain the above described SCB order-of-arrival priority semantic required for the proper operation of Young (i.e., "SCBs are appended to the common and target queues in order of arrival from host system 206"; col. 6, lines 25-27). This ordering is required by Young so that "priority of execution is maintained according to the order of delivery by the host system 206"; col. 8, lines 25-34. More particularly, component 255 of Fig. 3A merely describes a two-dimensional SCSI control block array, which has already been discussed above with respect to component **260A** of Fig. 3B. Additionally, Fig. 3C merely illustrates "the two-dimensional hardware command block execution queue of Fig. 3B" (column 3, lines 50-54). Fig. 6 provides another exemplary illustration of the already discussed "two-dimensional hardware command block execution queue" (column 3, lines 63-65). Moreover, Young at

column 5, line 46 or column 6, lines 7, merely describes operations to insert a control block into a two-dimensional queue of linked lists. Specifically, Young indicates that a control block is inserted into a first dimension when no other control block for that particular target exists in the first dimension, and that the **control block is always inserted into the second dimension when a control block for the target already exists in the first dimension**. This allows Young to maintain deliver each SCB in the order that it was received from an initiator, to the indicated target.

In view of the above, the Final Action's statement in sections 11 and 12 do not modify Young to show why each and every element of claim 1 would have been obvious to a person of ordinary skill in the art at the time of invention over AAPA and Young. This is especially the case since AAPA only teaches single-dimension queue management. AAPA will never insert anything into a first dimension based on analysis of anything in a second dimension. Additionally, Young will never result in an SCB for a target **not** being appended to the end of a target queue when there was already a different SCB in a common queue associated with the target queue for the target would destroy. As such, the combination of AAPA and Young still does not teach or suggest those features of claim 1 that the Final Action admits are missing. Thus, a system of AAPA and Young will never "responsive to the identifying, inserting the thread of execution into a first dimension of the multi-dimensional sleep queue if: [...] (b) if there are one or more different threads of execution with the wake-up time in a second dimension of the multi-dimensional sleep queue, each of the one [or] more

different threads of execution has a thread priority lower than or equal to a thread priority associated with the thread of execution", as claim 1 recites.

Accordingly, the 35 USC §103(a) rejection of claim 1 over AAPA in view of Young is improper and should be withdrawn.

As an additional matter, assuming that the Examiner was attempting to modify Young in sections 11 and 12 of the Final Action to arrive at the admitted missing features of claim 1, a factor cutting against any finding of motivation to modify prior art references to arrive at a claimed feature is when the prior art teaches away from the proposed modification. Appellant respectfully submits that any modification to AAPA and Young that would result in a SCB for a target not being appended to the end of a target queue when there was already a different SCB in a common queue associated with the target queue for the target would destroy Young's SCB order of arrival queue storage, which Young requires to maintain priority of SCB execution "according to the order of delivery by the host system" (col. 8, lines 25-34). Thus Young teaches away from any such modification. This cuts against any finding that a person of ordinary skill in the art at the time of invention would have had motivation to modify AAPA and Young to arrive at the features of claim 1. For these additional reasons, AAPA and Young do not teach or suggest the features of claim 1.

Claims 2-7, 9, 11, and 12 depend from claim 1 and are allowable over the cited combination solely by virtue of this dependency. For this reason alone, the 35 USC §103(a) rejection of claims 2-7, 9, 11, and 12 is improper and should be withdrawn.

Additionally, claims 7, 11, and 12 include additional features that are not taught or suggested by the cited combination of references.

For example, claim 7 recites "sorting the thread of execution into the first dimension based on respective thread wake-up times", "sorting the thread of execution into the second dimension based on respective thread priorities", and "wherein the thread of execution is sorted first with respect to the first dimension and second with respect to the second dimension." For the reasons already discussed above with respect to claim 1, AAPA and Young do not teach or suggest these claimed features. Moreover, a person or ordinary skill in the art at the time of invention would not have been motivated to modify AAPA and Young to arrive at these recited features.

Accordingly, and for this additional reason, the 35 USC §103(a) rejection of claim 7 should be withdrawn.

In another example, claim 11 recites in part "identifying a different thread in the first dimension that has a wake-up time equivalent to the predetermined amount of time", and "responsive to identifying the different thread: concluding that a first priority corresponding to the thread of execution is higher than a second priority corresponding to the different thread", and "replacing the different thread in the first dimension with the thread of execution, such the thread of execution is a member of both the first and the second dimensions, and such that the replaced thread has a secondary position with respect to the first and second dimensions." For the reasons already discussed above with respect to claim 1, AAPA and Young do not teach or suggest these claimed features. Moreover, a person or

ordinary skill in the art at the time of invention would not have been motivated to modify AAPA and Young to arrive at these recited features.

Accordingly, and for these additional reasons, the 35 USC §103(a) rejection of claim 11 should be withdrawn.

In yet another example, **claim 12** recites in part "identifying a different thread in the first dimension that has a wake-up time equivalent to the predetermined amount of time", and "responsive to identifying the different thread: determining that a first priority corresponding to the thread of execution is lower than a second priority that corresponds to the different thread", and "inserting the thread of execution into the second dimension, such the thread of execution occupies a secondary position with respect to the first and second dimensions and such that any different threads in the second dimension with lower priority than the first priority are subsequent in position to the secondary position."

For the reasons already discussed, AAPA does not teach or suggest a second dimension and Young merely maintains SCB order of arrival positioning in a queue—any other positioning would change the principal of operation of Young. Thus, AAPA in view of Young are not sufficient to render the claims prima facie obvious. The cited combination may never "inserting the thread of execution into the second dimension, such the thread of execution occupies a secondary position with respect to the first and second dimensions and such that any different threads in the second dimension with lower priority than the first priority are subsequent in position to the secondary position", claim 12 recites.

Accordingly, and for these additional reasons, the 35 USC §103(a) rejection of claim 12 should be withdrawn.

Claim 13 recites in part "inserting the thread of execution into a first dimension of the multi-dimensional sleep queue if: (a) there is not a thread with a wake-up time equivalent to the predetermined amount of time in the first dimension", and "(b) if there are one or more different threads of execution with the wake-up time in a second dimension of the multi-dimensional sleep queue, each of the one [or] more different threads of execution has a thread priority lower than or equal to a thread priority associated with the thread of execution." For the reasons already described above, AAPA in view of Young does not teach or suggest these claimed features.

Accordingly, for this reason alone, the 35 USC §103(a) rejection of claim 13 is improper and should be withdrawn.

Claims 14-18, and 20-23 depend from claim 13 and are allowable over the cited combination solely by virtue of this dependency. For this reason alone, the 35 USC §103(a) rejection of claims 14-18 and 20-23 is improper and should be withdrawn.

Additionally, for the reasons already described above, claims 18 and 22-23 include additional features that are not taught or suggested by the cited combination of references. For these additional reasons, the 35 USC §103(a) rejection of claims 18, and 22-23 should be withdrawn.

Claim 24 recites in part "inserting a new thread into the multi-dimensional sleep queue using a multi-dimensional atomic walk procedure". The Action concedes that AAPA is completely silent with respect to the claimed "multi-dimensional sleep queue", as claim 24 recites. However, the Action, at page 7, section 23, asserts that this claimed feature is taught by AAPA in view of Young

at col. 5, line 42 through column 6, line 18, and Fig. 4. This conclusion is unsupportable.

Young at col. 5, line 42 through col. 6, line 18, and referring to Fig. 4, explicitly describes:

> *"When there is a new SCB in SCB array 255, firmware implementing append operation 400 (FIG. 4) and executing on sequencer 225 reads the target number from the SCB in read target number operation 401, and processing transfers to valid tail pointer check operation 402. In operation 402, the target number is used as index into list 350 of target tail pointers. If the target tail pointer at the indexed location in list 350 has an invalid value, processing transfers to read common tail pointer operation 405 and otherwise to update SCB operation 403.*
>
> *If a valid target tail pointer exists for a target device, there is a target queue for that target in SCB array 255. Consequently, update SCB operation 403 accesses the SCB addressed by the target tail pointer read in operation 401. Update SCB operation 403 writes the location of the new SCB within SCB array 255 in field TQNEXT of the SCB addressed by the target tail pointer. Update SCB operation 403 transfers processing to update target tail pointer operation 404.*
>
> *In update target tail pointer operation 404, the target tail pointer in list 350 that is indexed by the target number in the new SCB is changed to point at the storage location in SCB array 255 of the new SCB. Operation 404 transfers to done operation 410, because the new SCB has been appended to the appropriate target queue and consequently two-dimensional queue 260 A.*
>
> *If a valid tail pointer does not exist for a target device in list 350, there is not an SCB for the target device in common queue 275. Consequently, read common tail pointer operation 405 reads the common tail pointer in memory 204 to determine the storage location of the last SCB in common queue 275. Read common tail pointer operation 405 transfers processing to update SCB operation 406.*
>
> *Update SCB operation 406 accesses the SCB addressed by common tail pointer. Update SCB operation 406 writes the location of the*

new SCB within SCB array 255 in field CQNEXT of the SCB addressed by common tail pointer. Update SCB operation 406 transfers processing to update target tail pointer operation 407."

Clearly, these quoted teachings of Young to use a tail pointer to determine if a target queue exists for a target device, update a tail pointer value, write to a data field of an SCB addressed by the tail pointer, and determine a storage location of a last SCB in the common queue, do not teach or suggest the claimed "multi-dimensional atomic walk procedure" of claim 24. In addressing this claimed feature, which as shown above is clearly missing from the cited portion of Young, the Final Action in section 37 asserts that "limitations of the specification cannot be read into the claims to avoid prior art." Appellant respectfully submits that limitations from the specification do not need to be read into the claimed "multi-dimensional atomic walk procedure" of claim 24 to avoid prior art. This is especially the case since the above cited portion, and Young as a whole, is completely silent with respect to "a multidimensional atomic walk" anything.

Accordingly, the 35 USC §103(a) rejection of claim 24 is improper and should be withdrawn.

Claims 25-30 depend from claim 24 and are allowable over the cited combination by virtue of this dependency. Accordingly, the 35 USC §103(a) rejection of claims 25-30 should be withdrawn.

Additionally, claim 30 includes additional features that are not taught or suggested by the cited combination of references. Claim 30 recites in part "determining if a status of a last examined thread has changed, the status indicating either that the last examined thread was removed from the multi-dimensional sleep queue, or indicating that the last examined thread was moved

from a first dimension of threads that is sorted based on respective thread wake-up times, to a second dimension of threads that is ordered based on respective thread priorities", "if the status of the last examined thread has changed, searching for the thread insertion point from a beginning of the multidimensional sleep queue", and "if the status of the last examined thread has not changed, searching for the thread insertion point from the last examined thread." Nowhere do the references of record teach or suggest these recited features.

In addressing this feature, the Action at page 29 concedes that "AAPA as modified does not specifically teach determining a status of a last examined thread and searching for thread insertion point to insert threads based on the status." To supply this missing feature, the Action points to Figs. 3A and 4, components 350, and 404-408 of Young, for showing a scratch memory that stores the value of tail pointers that points to the last SCB in a queue. The Action concludes that it "would have been obvious to one of ordinary skill in the art, to have recognized that changes in the status of the last examined thread can be used to determining a starting point for insertion because doing so will yield a more optimized insertion procedure by not having to traverse the entire queue to locate an insertion point if the status of a last examined node does not change. Appellant respectfully disagrees.

Patents are references only for what they clearly disclose or suggest and cannot be modified in structure to one that which it does not suggest. Young does not suggest the Final Action's modification (i.e., modifying Young so that one would "not having to traverse the entire queue to locate an insertion point"). Young does not suggest any such SCB queue traversal to identify insertion points

for an SCB that are any different than those explicitly described by Young. Implementing such traversal to locate such different positions for SCB insertion into the queue of Young would destroy the order-of-arrival SCB storage semantic of Young (i.e., SCBs would not be delivered to a target for execution in the same order that they were received from the host). For at least these reasons, Young does not traverse any queue to determine an insertion point for an SCB, nor does Young provide any suggestion to do so. Thus, the Final Action has failed to make a prima facie case of obviousness of claim 30 over the cited combination.

For these additional reasons, the 35 USC §103(a) rejection of claim 30 over the cited combination is improper and should be withdrawn.

Claim 31 recites "inserting a new thread into the sleep queue using a multi-dimensional atomic walk procedure", and "removing the new thread from the sleep queue for insertion into a run queue." For the reasons already discussed above with respect to claim 24, the cited combination of AAPA in view of Young does not teach or suggest these claimed features.

Accordingly, the 35 USC §103(a) rejection of claim 31 is improper and should be withdrawn.

Claims 32-37 depend from claim 31 and are allowable over the cited combination solely by virtue of this dependency. For this reason alone, the 35 USC §103(a) rejection of claims 32-38 is improper and should be withdrawn.

Additionally, for the reasons already described above with respect to claims 30, claim 37 includes additional features that are not taught or suggested by the cited combination of references. For these additional reasons, the 35 USC §103(a) rejection of claim 37 should be withdrawn.

Claim 39 recites "inserting the thread of execution into a first dimension of the multi-dimensional sleep queue if: (a) there is not a thread with a wake-up time equivalent to the predetermined amount of time in the first dimension", and "(b) if there are one or more different threads of execution with the wake-up time in a second dimension of the multi-dimensional sleep queue, each of the one [or] more different threads of execution has a thread priority lower than or equal to a thread priority associated with the thread of execution." This claim is rejected for the same rational used by the Action to reject claim 1. For the reasons already discussed with respect to claim 1, AAPA in view of Young does not teach or suggest these claimed features.

Accordingly, the 35 USC §103(a) rejection of claim 39 is improper and should be withdrawn.

Claims 40-44, 46, and 48-49 depend from claim 39 and are allowable over the cited combination solely by virtue of this dependency. For this reason alone, the 35 USC §103(a) rejection of claims 40-44, 46, and 48-49 is improper and should be withdrawn.

Additionally, for the reasons already described above with respect to claims 7, 11, and 12, claims 44, 48, and 49 include additional features that are not taught or suggested by the cited combination of references. For these additional reasons, the 35 USC §103(a) rejection of claims 44, 48, and 49 should be withdrawn.

Conclusion

Appellant respectfully submits that the rejections to the pending claims have been traversed. The pending claims are in condition for allowance and action to that end is urgently requested.

Respectfully Submitted,

Dated: 9/28/2005

By: Brian Hart

Brian G. Hart
Lee & Hayes, PLLC
Reg. No. 44,421
(509) 324-9256

APPENDIX OF APPEALED CLAIMS

1. (Previously presented) A computer implemented method for managing a multi-dimensional sleep queue, the method comprising:

identifying a thread of execution to insert into a sleep queue for a predetermined amount of time;

responsive to the identifying, inserting the thread of execution into a first dimension of the multi-dimensional sleep queue if:

(a) there is not a thread with a wake-up time equivalent to the predetermined amount of time in the first dimension; and

(b) if there are one or more different threads of execution with the wake-up time in a second dimension of the multi-dimensional sleep queue, each of the one ore more different threads of execution has a thread priority lower than or equal to a thread priority associated with the thread of execution.

2. (Previously presented) A computer implemented method as recited in claim 1, wherein the multi-dimensional sleep queue is a real-time multi-dimensional sleep queue.

3. (Previously presented) A computer implemented method as recited in claim 1, wherein the multi-dimensional sleep queue is a two-dimensional sleep queue.

4. (Previously presented) A computer implemented method as recited in claim 1, wherein inserting the thread of execution into the multi-dimensional sleep queue is performed in a manner that allows the thread scheduling mechanism to schedule other threads for execution within a deterministic amount of time.

5. (Previously presented) A computer implemented method as recited in claim 1, wherein inserting the thread of execution into the multi-dimensional sleep queue further comprises inserting the thread of execution into the multi-dimensional sleep queue such that a group of threads can be removed from the multi-dimensional sleep queue in a deterministic amount of time.

6. (Previously presented) A computer implemented method as recited in claim 1, wherein the multi-dimensional sleep queue comprises a group of threads, and wherein the method further comprises:

removing the group of threads from the multi-dimensional sleep queue in a deterministic amount of time, each thread in the group having a same wake-up time.

7. (Previously presented) A computer implemented method as recited in claim 1:

wherein the predetermined amount of time is a wake-up time, and wherein the thread of execution has a priority; and

wherein inserting the thread of execution into the multi-dimensional sleep queue further comprises

sorting the thread of execution into the first dimension based on respective thread wake-up times; and

sorting the thread of execution into the second dimension based on respective thread priorities; and

wherein the thread of execution is sorted first with respect to the first dimension and second with respect to the second dimension.

8. (Canceled).

9. (Previously presented) A computer implemented method as recited in claim 1, wherein the second dimension comprises a plurality of threads, each thread of the threads having a same respective thread wake-up time.

10. (Canceled).

11. (Previously presented) A computer implemented method as recited in claim 7, wherein sorting the thread of execution further comprises:

identifying a different thread in the first dimension that has a wake-up time equivalent to the predetermined amount of time; and

responsive to identifying the different thread:

concluding that a first priority corresponding to the thread of execution is higher than a second priority corresponding to the different thread; and

replacing the different thread in the first dimension with the thread of execution, such the thread of execution is a member of both the first and the second dimensions, and such that the replaced thread has a secondary position with respect to the first and second dimensions.

12. (Previously presented) A computer implemented method as recited in claim 7, wherein sorting the thread of execution further comprises:

identifying a different thread in the first dimension that has a wake-up time equivalent to the predetermined amount of time; and

responsive to identifying the different thread:

determining that a first priority corresponding to the thread of execution is lower than a second priority that corresponds to the different thread; and

inserting the thread of execution into the second dimension, such the thread of execution occupies a secondary position with respect to the first and second dimensions and such that any different thread in the second dimension with lower priority than the first priority is subsequent in position to the secondary position.

13. (Previously presented) A computer-readable medium for managing a multi-dimensional sleep queue, the computer-readable medium comprising computer-executable instructions for:

identifying a thread of execution to insert into a sleep queue for a predetermined amount of time;

responsive to the identifying, inserting the thread of execution into a first dimension of the multi-dimensional sleep queue if:

(a) there is not a thread with a wake-up time equivalent to the predetermined amount of time in the first dimension; and

(b) if there are one or more different threads of execution with the wake-up time in a second dimension of the multi-dimensional sleep queue, each of the one ore more different threads of execution has a thread priority lower than or equal to a thread priority associated with the thread of execution.

14. (Original) A computer-readable medium as recited in claim 13, wherein the multi-dimensional sleep queue is a real-time multi-dimensional sleep queue.

15. (Previously presented) A computer-readable medium as recited in claim 13, wherein the computer-executable instructions for inserting the thread of execution into the multi-dimensional sleep queue are performed in a manner that allows the thread scheduling mechanism to schedule other threads for execution in a deterministic amount of time.

16. (Previously presented) A computer-readable medium as recited in claim 13, wherein the computer-executable instructions for inserting the thread of execution into the multi-dimensional sleep queue further comprise instructions for inserting the thread of execution into the multi-dimensional sleep queue such that a group of threads can be removed from the multi-dimensional sleep queue in a deterministic amount of time.

17. (Original) A computer-readable medium as recited in claim 13, wherein the multi-dimensional sleep queue comprises a group of threads, and wherein the computer-executable instructions further comprise instructions for:

removing the group of threads from the multi-dimensional sleep queue in a deterministic amount of time, each thread in the group having a same wake-up time.

18. (Previously presented) A computer-readable medium as recited in claim 13, wherein the predetermined amount of time is a wake-up time, and wherein the thread of execution has a priority; and

wherein the computer-executable instructions for inserting the thread of execution into the multi-dimensional sleep queue further comprise instructions for:

sorting the thread of execution into the second dimension based on respective thread priorities; and

wherein the thread of execution is sorted first with respect to the first dimension and second with respect to the second dimension.

19. (Canceled).

20. (Previously presented) A computer-readable medium as recited in claim 13, wherein the second dimension comprises a plurality of threads, each thread of the threads having a same respective thread wake-up time.

21. (Previously presented) A computer-readable medium as recited in claim 13, wherein the thread of execution is a new thread, and wherein the instructions for inserting the thread of execution further comprise instructions for:

determining that the predetermined amount of time is different as compared to each respective wake-up time of each other thread in the first dimension ; and

responsive to the determining, introducing the new thread into the first dimension.

22. (Previously presented) A computer-readable medium as recited in claim 18, wherein the computer-executable instructions for sorting the thread further comprise instructions for:

identifying a different thread in the first dimension that has a wake-up time equivalent to the predetermined amount of time; and

responsive to identifying the different thread:

concluding that a first priority corresponding to the thread of execution is higher than a second priority corresponding to the different thread; and

replacing the different thread in the first dimension with the thread of execution, such the thread of execution is a member of both the first and the second dimensions, and such that the replaced thread has a secondary position with respect to the first and second dimensions.

23. (Previously presented) A computer-readable medium as recited in claim 18, wherein the computer-executable instructions for sorting the thread further comprise instructions for:

identifying a different thread in the first dimension that has a wake-up time equivalent to the predetermined amount of time; and

responsive to identifying the different thread:

determining that a first priority corresponding to the thread of execution is lower than a second priority that corresponds to the different thread; and

inserting the thread of execution into the second dimension, such the thread of execution occupies a secondary position with respect to the first and second dimensions and such that any different thread in the second dimension with lower priority than the first priority is subsequent in position to the secondary position.

24. (Previously presented) A computer implemented method for managing a multi-dimensional sleep queue comprising:

inserting a new thread into the multi-dimensional sleep queue using a multi-dimensional atomic walk procedure; and

removing the new thread from the multi-dimensional sleep queue for insertion into a run queue.

25. (Previously presented) A computer implemented method as recited in claim 24, wherein inserting the new thread further comprises:

if the new thread is a first thread, setting a last examined thread to reference the new thread, the last examined thread being used to identify an insertion point for the new thread.

26. (Previously presented) A computer implemented method as recited in claim 24, further comprising:

removing a group of threads from the multi-dimensional sleep queue in a deterministic amount of time.

27. (Previously presented) A computer implemented method as recited in claim 24, further comprising:

removing a group of threads from the sleep queue in a deterministic amount of time, each thread in the group of threads having a same wake-up time.

28. (Previously presented) A computer implemented method as recited in claim 27, wherein the deterministic amount of time is independent of a number of threads in the group of threads.

29. (Previously presented) A computer implemented method as recited in claim 24, wherein the multi-dimensional sleep queue comprises at least one other thread, and wherein inserting the new thread further comprises:

establishing a thread insertion point in the multi-dimensional sleep queue for the new thread; and

introducing the new thread into the multi-dimensional sleep queue at the insertion point.

30. (Previously presented) A computer implemented method as recited in claim 29, wherein establishing the thread insertion point further comprises:

determining if a status of a last examined thread has changed, the status indicating either that the last examined thread was removed from the multi-dimensional sleep queue, or indicating that the last examined thread was moved from a first dimension of threads that is sorted based on respective thread wake-up times, to a second dimension of threads that is ordered based on respective thread priorities;

if the status of the last examined thread has changed, searching for the thread insertion point from a beginning of the multidimensional sleep queue; and

if the status of the last examined thread has not changed, searching for the thread insertion point from the last examined thread.

31. (Original) A computer-readable medium comprising computer-executable instructions for managing a sleep queue, the computer executable instructions comprising instructions for:

inserting a new thread into the sleep queue using a multi-dimensional atomic walk procedure; and

removing the new thread from the sleep queue for insertion into a run queue.

32. (Original) A computer-readable medium as recited in claim 31, wherein instructions for inserting the new thread further comprise instructions for:

if the new thread is a first thread, setting a last examined thread to reference the new thread, the last examined thread being used to identify an insertion point for the new thread.

33. (Original) A computer-readable medium as recited in claim 31, further comprising instructions for:

removing a group of threads from the sleep queue in a deterministic amount of time.

34. (Original) A computer-readable medium as recited in claim 31, further comprising instructions for:

removing a group of threads from the sleep queue in a deterministic amount of time, each thread in the group of threads having a same wake-up time.

35. (Original) A computer-readable medium as recited in claim 34, wherein the deterministic amount of time is independent of a number of threads in the group of threads.

36. (Original) A computer-readable medium as recited in claim 31, wherein the multi-dimensional sleep queue comprises at least one other thread, and wherein the instructions for inserting the new thread further comprise instructions for:

establishing a thread insertion point in the multi-dimensional sleep queue for the new thread; and

introducing the new thread into the multi-dimensional sleep queue at the insertion point.

37. (Previously presented) A computer-readable medium as recited in claim 36, wherein the instructions for establishing the thread position further comprise instructions for:

determining if a status of a last examined thread has changed, the status indicating either that the last examined thread was removed from the multi-dimensional sleep queue, or indicating that the last examined thread was moved from a first dimension of threads that is sorted based on respective thread wake-up times, to a second dimension of threads that is ordered based on respective thread priorities;

if the status of the last examined thread has changed, searching for the thread insertion point from a beginning of the multidimensional sleep queue; and

if the status of the last examined thread has not changed, searching for the thread insertion point from the last examined thread.

38. (Canceled).

39. (Previously presented) A system for managing a sleep queue, the system comprising:

a memory comprising computer-executable instructions and a multi-dimensional sleep queue; and

a processor operatively coupled to the memory for executing the computer-executable instructions, the computer-executable instructions comprising instructions for:

identifying a thread of execution to insert into a sleep queue for a predetermined amount of time;

responsive to the identifying, inserting the thread of execution into a first dimension of the multi-dimensional sleep queue if:

(a) there is not a thread with a wake-up time equivalent to the predetermined amount of time in the first dimension; and

(b) if there are one or more different threads of execution with the wake-up time in a second dimension of the multi-dimensional sleep queue, each of the one ore more different threads of execution has a thread priority lower than or equal to a thread priority associated with the thread of execution.

40. (Original) A system as recited in claim 39, wherein the multi-dimensional sleep queue is a real-time multi-dimensional sleep queue.

41. (Previously presented) A system as recited in claim 39, wherein the instructions for inserting the thread of execution are performed in a manner that allows a thread scheduling mechanism to schedule other threads for execution in a deterministic amount of time.

42. (Previously presented) A system as recited in claim 39, wherein the instructions for inserting the thread of execution further comprise instructions for:

inserting the thread of execution into the multi-dimensional sleep queue such that a group of threads can be removed from the multi-dimensional sleep queue in a deterministic amount of time.

43. (Original) A system as recited in claim 39, wherein the multi-dimensional sleep queue further comprises a group of threads, and wherein the computer executable instructions further comprise instructions for:

inserting the group of threads into the multi-dimensional sleep queue; and

removing the group of threads from the multi-dimensional sleep queue in a deterministic amount of time, each thread in the group having a same wake-up time.

44. (Previously presented) A system as recited in claim 39, wherein the thread has a wake-up time and a priority, and wherein the instructions for inserting the thread further comprise instructions for:

sorting the thread of execution into the first dimension based on respective thread wake-up times; and

sorting the thread of execution into the second dimension based on respective thread priorities; and

wherein the thread of execution is sorted first with respect to the first dimension and second with respect to the second dimension.

45. (Canceled).

46. (Previously presented) A system as recited in claim 44, wherein the second dimension comprises a plurality of threads, each thread of the threads having a same respective thread wake-up time.

47. (Canceled).

48. (Previously presented) A system as recited in claim 44, wherein the instructions for sorting the thread of execution further comprise instructions for:

identifying a different thread in the first dimension that has a wake-up time equivalent to the predetermined amount of time; and

responsive to identifying the different thread:

concluding that a first priority corresponding to the thread of execution is higher than a second priority corresponding to the different thread; and

replacing the different thread in the first dimension with the thread of execution, such the thread of execution is a member of both the first and the second dimensions, and such that the replaced thread has a secondary position with respect to the first and second dimensions.

49. (Previously presented) A system as recited in claim 44, wherein the instructions for sorting the thread of execution further comprise instructions for:

identifying a different thread in the first dimension that has a wake-up time equivalent to the predetermined amount of time; and

responsive to identifying the different thread:

determining that a first priority corresponding to the thread of execution is lower than a second priority that corresponds to the different thread; and

inserting the thread of execution into the second dimension, such the thread of execution occupies a secondary position with respect to the first and second dimensions and such that any different thread in the second dimension with lower priority than the first priority is subsequent in position to the secondary position.

50-56. (Canceled).